Listing of Claims:

1. (Currently amended) A method comprising:

determining, during a process of booting a processing system, whether sufficient resources are available to meet resource requests for multiple devices associated with a peripheral connect interface (PCI) subsystem of the processing system;

retrieving boot information from an extended firmware interface (EFI) environment of the processing system;

automatically identifying at least one of the multiple devices as boot-critical, based on the boot information from the EFI environment;

automatically determining whether the at least one boot-critical device connects via an intermediary device in the processing system;

if the at least one boot-critical device connects via the intermediary device, automatically identifying the intermediary device as a boot critical device; and

if sufficient resources to meet the resource requests are not available, automatically allocating resources for ~~the device~~ devices identified as boot-critical but not for all of the devices associated with the PCI subsystem, before the processing system boots an operating system (OS).

2. (Original) A method according to claim 1, further comprising:

identifying at least one device associated with the PCI subsystem as not critical to booting the processing system; and

disabling the device identified as not critical, before the processing system boots the OS, if sufficient resources to meet the resource requests are not available.

3. (Original) A method according to claim 1, wherein the operation of retrieving boot information from an EFI environment of the processing system comprises:

retrieving information that designates at least one device selected from a group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

4. (Original) A method according to claim 1, further comprising:

discovering the devices associated with the PCI subsystem by scanning the PCI subsystem.

5. (Original) A method according to claim 1, further comprising:

collecting resource requests from at least two of the devices associated with the PCI subsystem; and

determining whether sufficient resources are available, based on the collected resource requests.

6. (Original) A method according to claim 1, further comprising:

determining available resources of the processing system, based on resource information from a host bridge driver.

7. (Original) A method according to claim 6, wherein the operation of determining whether sufficient resources are available to meet the resource requests comprises:

automatically determining whether sufficient resources are available, based on the resource information from the host bridge driver.

8. (Original) A method according to claim 1, further comprising:

automatically identifying at least one of the devices associated with the PCI subsystem to be rejected, based on a rejection policy.

9. (Original) A method according to claim 1, further comprising:

collecting resource requests from at least two of the devices associated with the PCI subsystem;

determining available resources of the processing system, based on resource information from a host bridge driver.

automatically determining whether sufficient resources are available for the devices associated with the PCI subsystem, based on the collected resource requests and the resource information from the host bridge driver; and

disabling at least one of the devices associated with the PCI subsystem before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.

10. (Currently amended) An apparatus comprising:

a processor;

a peripheral connect interface (PCI) subsystem communicatively coupled to the processor;

two or more devices associated with the PCI subsystem;

a machine accessible medium communicatively coupled to the processor; and

a bus driver encoded in the machine accessible medium, wherein the bus driver, when executed by the processor, performs operations comprising:

determining, during a process of booting the processing system, whether sufficient resources are available to meet resource requests for the two or more devices;

retrieving boot information from an extended firmware interface (EFI) environment of the processing system;

automatically identifying at least one of the devices as boot-critical, based on the boot information from the EFI environment;

automatically determining whether the at least one boot-critical device connects via an intermediary device in the processing system;

4

<u>if the at least one boot-critical device connects via the intermediary device,</u> <u>automatically identifying the intermediary device as a boot critical device;</u> and

if sufficient resources to meet the resource requests are not available, facilitating successful booting of an operating system (OS) by automatically allocating resources for ~~the device~~ <u>devices</u> identified as boot-critical but not for all of the devices associated with the PCI subsystem, before the processing system boots the OS.

11. (Original) An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

identifying at least one device associated with the PCI subsystem as not critical to booting the processing system; and

disabling the device identified as not critical, before the processing system boots the OS, if sufficient resources to meet the resource requests are not available.

12. (Original) An apparatus according to claim 10, wherein the bus driver retrieves boot information from the EFI environment designating at least one device selected from a group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

13. (Original) An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

discovering the devices associated with the PCI subsystem by scanning the PCI subsystem.

14. (Original) An apparatus according to claim 10, wherein the bus driver performs further operations, comprising:

collecting resource requests from at least two of the devices associated with the PCI subsystem; and

determining whether sufficient resources are available, based on the collected resource requests.

15. (Original) An apparatus according to claim 10, wherein:

the apparatus further comprises a host bridge driver encoded in the machine accessible medium; and

the bus driver determines available resources of the processing system, based on resource information obtained from the host bridge driver.

16. (Original) An apparatus according to claim 15, wherein the bus driver automatically determines whether sufficient resources are available, based on the resource information obtained from the host bridge driver.

17. (Original) An apparatus according to claim 10, wherein, if sufficient resources are not available, the bus driver automatically identifies at least one of the devices associated with the PCI subsystem to be rejected, based on a rejection policy.

18. (Original) An apparatus according to claim 10, wherein:

the apparatus further comprises a host bridge driver encoded in the machine accessible medium; and

the operations performed by the bus driver further comprise:

collecting resource requests from at least two of the devices associated with the PCI subsystem;

determining available resources of the processing system, based on resource information from the host bridge driver;

automatically determining, based on the collected resource requests and the resource information from the host bridge driver, whether sufficient resources are available for the devices associated with the PCI subsystem; and

disabling at least one of the devices associated with the PCI subsystem, before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.

19. (Original) An apparatus comprising:

a tangible, machine accessible medium; and

instructions encoded in the machine accessible medium, wherein the instructions, when executed by a processing system, cause the processing system to perform operations comprising:

determining, during a process of booting the processing system, whether sufficient resources are available to meet resource requests for two or more devices associated with a peripheral connect interface (PCI) subsystem of the processing system;

retrieving boot information from an extended firmware interface (EFI) environment of the processing system;

automatically identifying at least one of the devices as boot-critical, based on the boot information from the EFI environment;

automatically determining whether the at least one boot-critical device connects via an intermediary device in the processing system;

7

if the at least one boot-critical device connects via the intermediary device, automatically identifying the intermediary device as a boot critical device; and

if sufficient resources are not available to meet the resource requests, automatically allocating resources for ~~the~~ boot-critical ~~device~~ devices but not for all of the devices associated with the PCI subsystem, before the processing system boots an operating system (OS).

20. (Original) An apparatus according to claim 19, further comprising:

a bus driver encoded in the machine accessible medium, wherein the bus driver comprises the instructions.

21. (Original) An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

identifying at least one device associated with the PCI subsystem as not critical to booting the processing system; and

disabling the device identified as not critical, before the processing system boots the OS, if sufficient resources to meet the resource requests are not available.

22. (Original) An apparatus according to claim 19, wherein the operation of retrieving boot information from an EFI environment of the processing system comprises:

retrieving information that designates at least one device selected from a group consisting of:

a device to serve as an input console during boot;

a device to serve as an output console during boot; and

a device to provide OS boot instructions.

23. (Original) An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

discovering the devices associated with the PCI subsystem by scanning the PCI subsystem.

24. (Original) An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

collecting resource requests from at least two of the devices associated with the PCI subsystem; and

determining whether sufficient resources are available, based on the collected resource requests.

25. (Original) An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

determining available resources of the processing system, based on resource information from a host bridge driver.

26. (Original) An apparatus according to claim 25, wherein the operation of determining whether sufficient resources are available comprises:

automatically determining whether sufficient resources are available, based on the resource information from the host bridge driver.

27. (Original) An apparatus according to claim 19, wherein:

if sufficient resources are not available, the instructions automatically identify at least one of the devices associated with the PCI subsystem to be rejected, based on a rejection policy.

28. (Original) An apparatus according to claim 19, wherein the operations performed by the instructions further comprise:

collecting resource requests from at least two of the devices associated with the PCI subsystem;

determining available resources of the processing system, based on resource information from a host bridge driver;

automatically determining, based on the collected resource requests and the resource information from the host bridge driver, whether sufficient resources are available for the devices associated with the PCI subsystem; and

disabling at least one of the devices associated with the PCI subsystem, before the processing system boots the OS, if sufficient resources are not available to meet the resource requests.

29. (New) A method according to claim 1, further comprising:

creating a device map to describe a topology for the devices associated with the PCI subsystem before the processing system boots the OS.

30. (New) A method according to claim 29, further comprising:

using the device map to automatically determine whether any boot critical devices connect via any intermediary devices in the processing system.

31. (New) A method according to claim 1, wherein:

the operation of retrieving boot information from the EFI environment comprises retrieving device identifiers for the devices associated with the PCI subsystem; and

the operation of automatically identifying at least one of the multiple devices as boot-critical is performed by a PCI bus driver, based on information in the PCI bus driver designating certain device identifiers as boot-critical.

32. (New) A method according to claim 31, wherein:

the operation of retrieving device identifiers from the EFI environment of the processing system comprises retrieving device path designations from at least three EFI boot device selection (BDS) variables from the group consisting of:

an EFI console-in variable;

an EFI console-out variable;

an EFI error-out variable; and

an EFI boot device variable; and

the device path designations are used to automatically identify at least one of the multiple devices as boot-critical.

33. (New) An apparatus according to claim 10, wherein the operations to be performed by the bus driver further comprise:

creating a device map to describe a topology for the devices associated with the PCI subsystem before the processing system boots the OS.

34. (New) An apparatus according to claim 33, wherein the operations to be performed by the bus driver further comprise:

using the device map to automatically determine whether any boot critical devices connect via any intermediary devices in the processing system.

35. (New) An apparatus according to claim 10, wherein:

the operation of retrieving boot information from the EFI environment comprises retrieving device identifiers for the devices associated with the PCI subsystem; and

the operation of automatically identifying at least one of the multiple devices as boot-critical is performed, based on information in the PCI bus driver designating certain device identifiers as boot-critical.

36. (New) An apparatus according to claim 35, wherein:

the operation of retrieving device identifiers from the EFI environment of the processing system comprises retrieving device path designations from at least three EFI boot device selection (BDS) variables from the group consisting of:

an EFI console-in variable;

an EFI console-out variable;

an EFI error-out variable; and

an EFI boot device variable; and

the device path designations are used to automatically identify at least one of the multiple devices as boot-critical.

37. (New) An apparatus according to claim 19, wherein the operations further comprise:

creating a device map to describe a topology for the devices associated with the PCI subsystem before the processing system boots the OS.

38. (New) An apparatus according to claim 37, wherein the operations further comprise:

using the device map to automatically determine whether any boot critical devices connect via any intermediary devices in the processing system.

39. (New) An apparatus according to claim 19, wherein:

the operation of retrieving boot information from the EFI environment comprises retrieving device identifiers for the devices associated with the PCI subsystem; and

the operation of automatically identifying at least one of the multiple devices as boot-critical is performed, based on information in the PCI bus driver designating certain device identifiers as boot-critical.

40. (New) An apparatus according to claim 39, wherein:

the operation of retrieving device identifiers from the EFI environment of the processing system comprises retrieving device path designations from at least three EFI boot device selection (BDS) variables from the group consisting of:

an EFI console-in variable;

an EFI console-out variable;

an EFI error-out variable; and

an EFI boot device variable; and

the device path designations are used to automatically identify at least one of the multiple devices as boot-critical.